Goals and vision for enabling advanced accelerator modeling under Compass

- Help in **designing linear colliders** based on staging PWFA or LWFA sections
- Develop **integrated codes** for modeling a staged wakefield "system".
- Develop **efficient and high fidelity modeling** for optimizing a single LWFA or PWFA stage.
- Develop **PIC algorithms for advanced architectures**
- Enable **routine modeling** of existing and future experiments:
  - BELLA
  - FACET
  - Others
  - Real time steering of experiments?
  - Code validation against numerous worldwide experiments
- **Code Verification**

UCLA

TECH

BERKELEY LAB

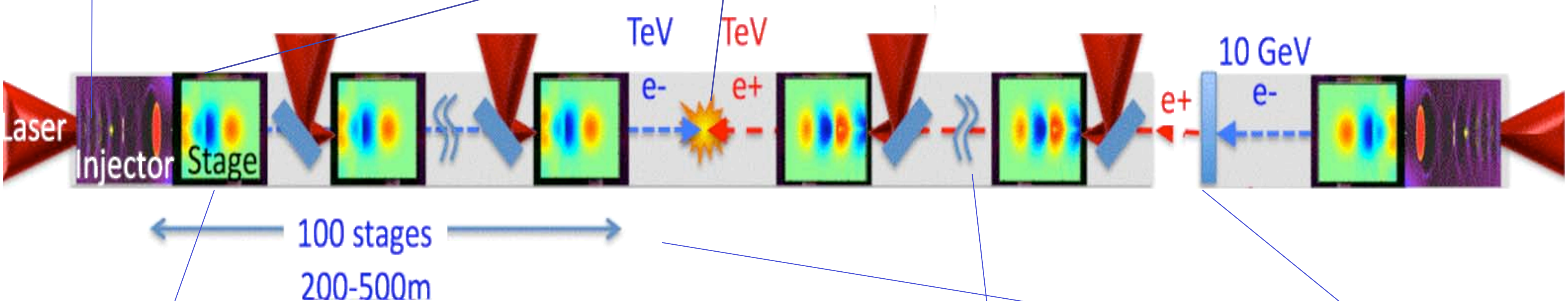
Office of Science

SciDAC
Scientific Discovery through Advanced Computing

Simulation needs for BELLA project and a laser – plasma collider

- 1: Low emittance injector
  - Downramp (Envelope,Explicit)
  - Colliding pulse (Explicit)

- 5: Bunch emittance
  - Accurate momentum advance
    (weighting, mesh refinement, QS PIC, high order models)
  - Noise control (fluids, EM dispersion, QS PIC, Cerenkov)



- 2: 10 GeV m-scale stages, PW laser
  - 1000x problem size (Scaling)
    Fast reduced models (Envelope, Quasi-static, Lorentz boosted)
  - Laser v_{group} (EM dispersion)
  - Error accumulation control
  - Hydro sim. of capillaries, jets

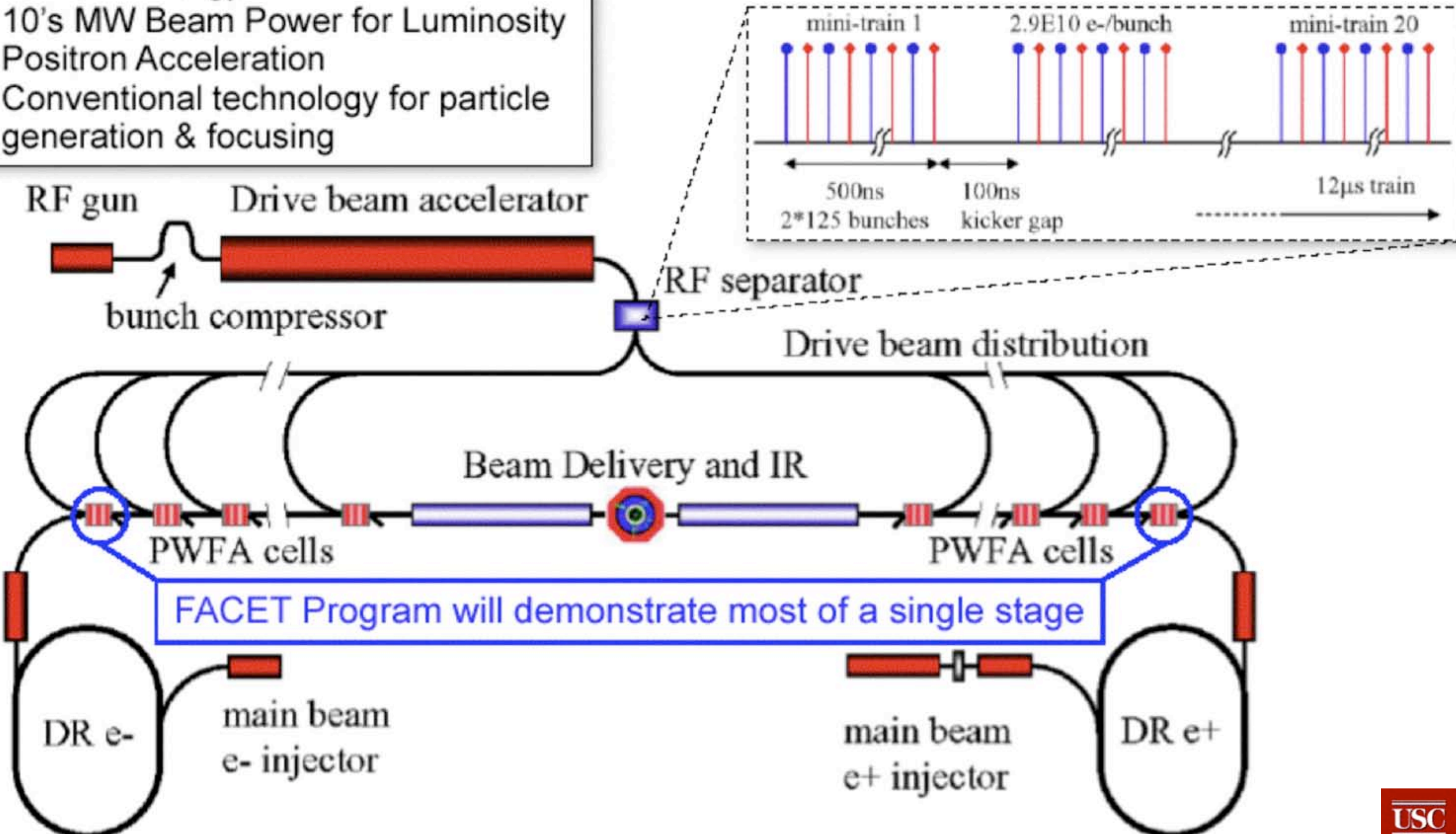
- 4: Scattering and radiation

- 3: Staging
  - Beam propagation
  - Plasma mirrors (Explicit)

BELLA will conduct experiments on these issues, requiring new modeling capability

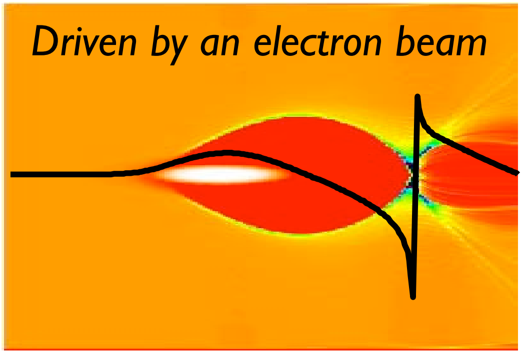
A concept for a Plasma Wakefield Accelerator based linear collider

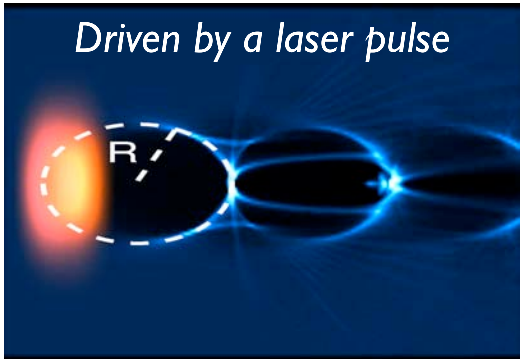
- TeV CM Energy
- 10's MW Beam Power for Luminosity
- Positron Acceleration
- Conventional technology for particle generation & focusing

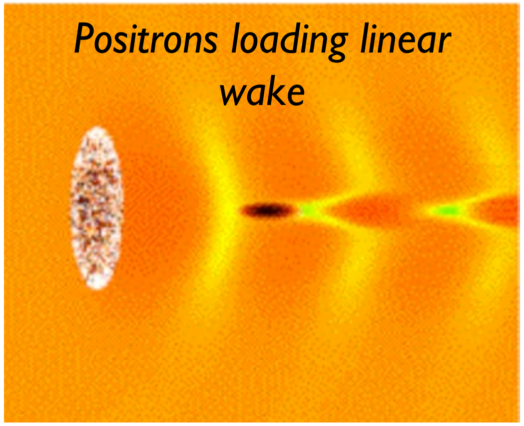


Plasma response of drive and/or trailing beam leads to the need of PIC code models

Rosenzweig et al. 1990, Lu et al. 2006





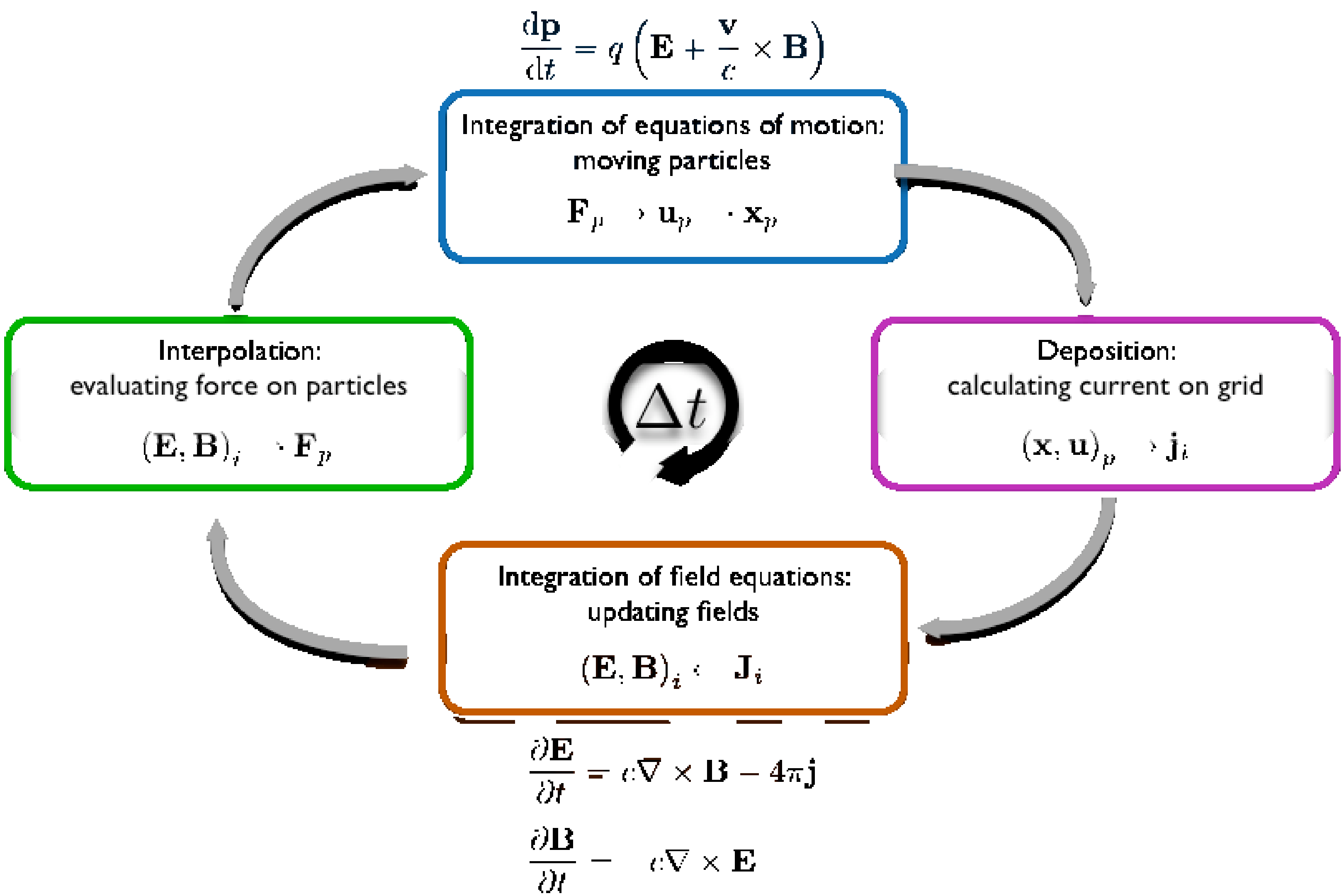


- Wakes in nonlinear regime necessitate particle methods
  - Ion channel formed by trajectory crossing
  - Fluid model breaks down
- Self-injection schemes require particle methods
- Beam loading in weakly nonlinear wakes require particle methods

What do we want to know and predict?

- Wake excitation for given drive beam ……
- Evolution of drive beam, e.g, instabilities…
- Transformer ratio, shaped bunches, train of bunches
- Beam loading, beam quality ……
- How to put these all together in a design?
- What about positrons?
- Effects of staging

Particle-in-cell algorithm
Not all PIC codes are the same!



Challenge in PIC modeling (will work on this)

Typical 3D high fidelity PWFA/LWFA simulation requirement

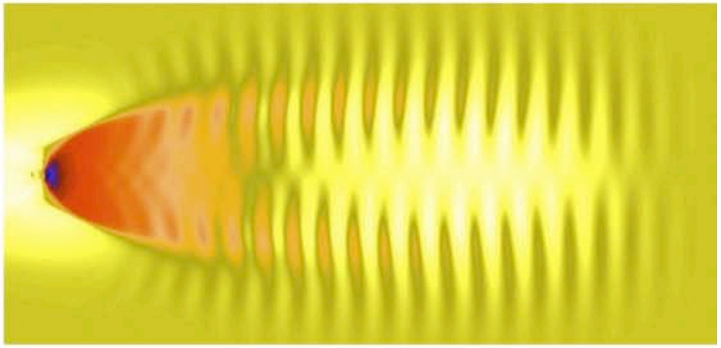
	Feature	Grid size limit	Time step limit	Total time of simulation per GeV stage (node-hour)*
PWFA	Full EM PIC	$\sim 0.05c/\omega_p$	$\Delta t < 0.05\omega_p^{-1}$	1500
	Quasi-static PIC	$\sim 0.05c/\omega_p$	$\Delta t < 0.05\omega_\beta^{-1} = 0.05 \times \sqrt{2\gamma}\omega_p^{-1}$	20 (4)

	Feature	Grid size limit	Time step limit	Total time of simulation per GeV stage (node-hour)
LWFA	Full EM PIC	$\sim 0.05\ \lambda$	$\Delta t < .25\omega_0^{-1}$	~ 500000 Using a Lorentz frame can reduce this
	Ponderomotive Guiding center PIC	$\sim 0.05c/\omega_p$	$\Delta t < 0.05\omega_p^{-1}$	~(1500)
	Quasi-static PIC	$\sim 0.05c/\omega_p$	$\Delta t < 0.05\ t_r$	~ 200 (10) No self-trapping but external injection

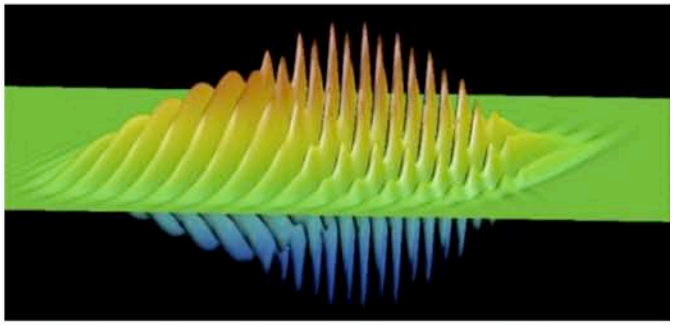
*These are rough estimates and represent potential speed up assuming particles dominate calculation. In some cases we have not reached the full potential. Resolution and particles per cell can impact these estimates

VORPAL provides state-of-the-art algorithms for laser-plasma simulations

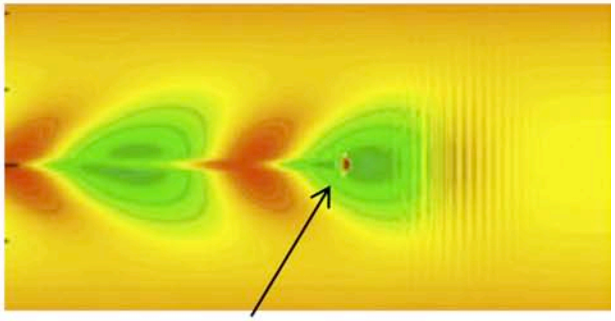
- Successfully applied to various LWFA problems:
  - C.G.R. Geddes *et al*., in SciDAC Review (2009), to appear.
  - K.J. Wu *et al*., in SciDAC Review (2009), to appear.
  - N. Hafz *et al*., Nature Photonics **2**, 571 (2008).
  - C. Geddes *et al*., Phys. Rev. Lett. **100**, 215004 (2008).
  - K. Nemeth *et al*., Phys. Rev. Lett. **100**, 095002 (2008).
  - J.R. Cary *et al*., Phys. Plasmas (2005), invited.
  - C. Geddes *et al.,* Nature **431**, 538 (2004).
  - C. Nieter & J.R. Cary, J. Comput. Phys. **196**, 448 (2004).
- Implemented algorithms include:
  - relativistic, electromagnetic time-explicit PIC and fluid
  - Lorentz-boosted simulations in 1D, 2D
  - Ponderomotive guiding center (PGC) PIC or “envelope” model
- Special features include:
  - High-order spline-based particle shapes (up through 5th)
  - PML (perfectly matched layer) absorbing boundaries
  - Tunneling-induced field ionization of H, He
- Parallel framework for particles and Cartesian meshes
  - Scales to >10,000 cores for production runs
  - Cross-platform (Linux, AIX, OS X, Windows)
  - 1D, 2D, 3D; combine algorithms at run-time
- VORPAL development team
  - about 30 developers; >10 active at any time
  - software version control; branching; nightly regression tests



3D Laser wakefield accelerator



Obliquely colliding laser pulses



Fluid/PIC; e- bunch in channel

http://www.txcorp.com/products/VORPAL

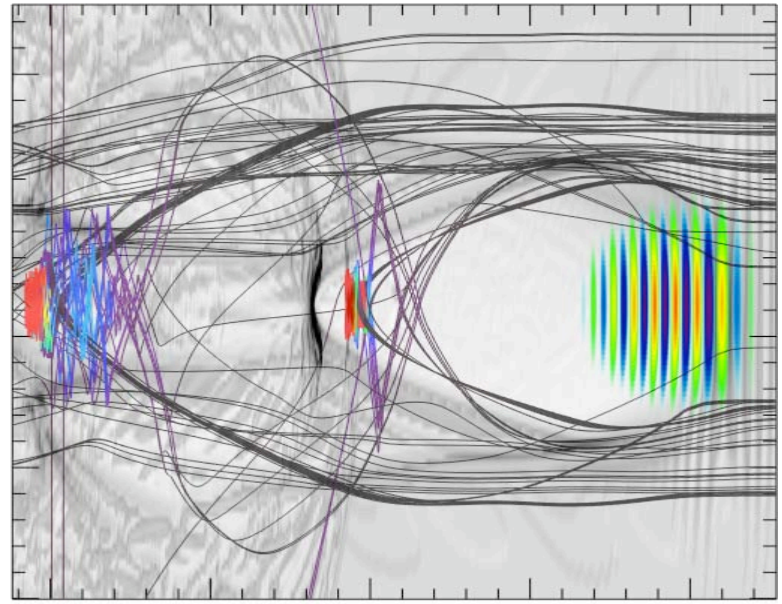
TECH-X CORPORATION

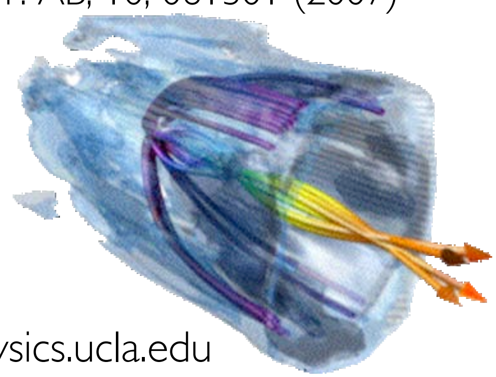
OSIRIS 2.0

osiris framework

- Massivelly Parallel, Fully Relativistic Particle-in-Cell (PIC) Code
- Visualization and Data Analysis Infrastructure
- Developed by the osiris.consortium: UCLA + IST
- Widely used: UCLA, SLAC, USC, Michigan, Rochester, IST, Imperial College, Max Planck Inst.

- Examples of applications
- Mangles et al., Nature 431 529 (2004).
- Tsung et al., Phys. Rev. Lett., 94 185002 (2004)
- Mangles et al., Phys. Rev. Lett., 96, 215001 (2006)
- Lu et al., Phys. Rev. ST: AB, 10, 061301 (2007)





New Features in v2.0

- Bessel Beams
- Binary Collision Module
- Tunnel (ADK) and Impact Ionization
- Dynamic Load Balancing
- PML absorbing BC
- Optimized Higher Order Splines
- Parallel I/O (HDF5)
- Boosted Frame in 1/2/3D

Frank Tsung: tsung@physics.ucla.edu
Ricardo Fonseca: ricardo.fonseca@ist.utl.pt
http://exodus.physics.ucla.edu
http://cfp.ist.utl.pt/golp/epp

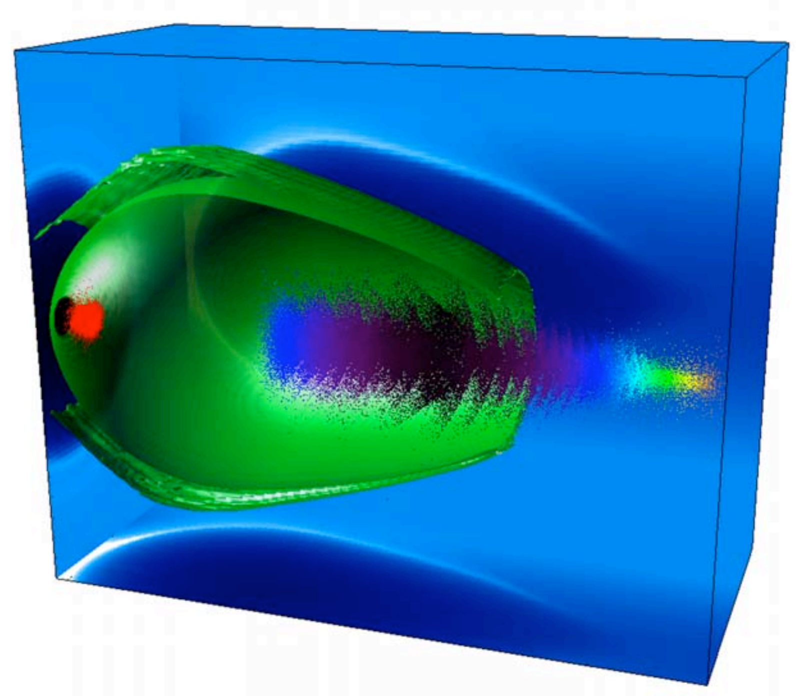
QuickPIC: 3D quasi-static particle code

Description

- Massivelly Parallel, 3D Quasi-static particle-in-cell code
- Ponderomotive guiding center for laser driver
- 100-1000+ savings with high fidelity
- Field ionization and radiation reaction included
- Simplified version used for e-cloud modeling
- Developed by UCLA + Umaryland + IST

Examples of applications

- Simulations for PWFA experiments, E157/162/164/164X/167 (Including Feb. 2007 Nature)
- Study of electron cloud effect in LHC.
- Plasma afterburner design up to TeV
- Efficient simulation of externally injected LWFA
- Beam loading studies using laser/beam drivers



New Features

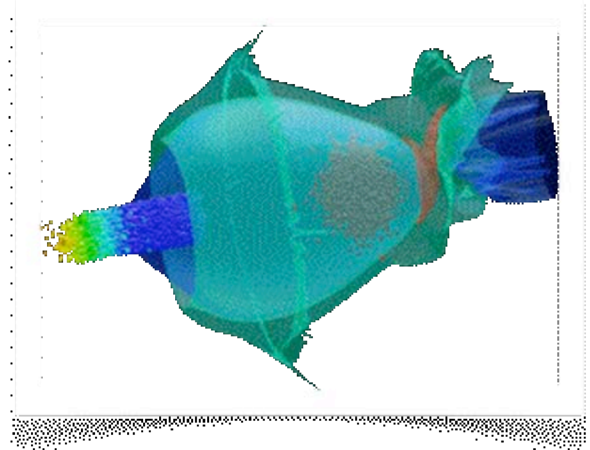
- Particle tracking
- Pipelining
- Parallel scaling to 1,000+ processors
- Beta version of enhanced pipelining algorithm: enables scaling to 10,000+ processors and unprecedented simulation resolution down to nm

Chengkun Huang:
huangck@ee.ucla.edu
http://exodus.physics.ucla.edu/
http://cfp.ist.utl.pt/golp/epp

UPIC: UCLA Particle-in-Cell Framework

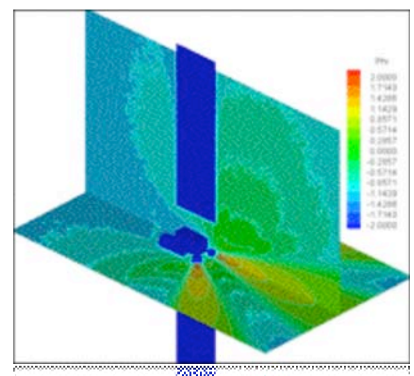
Main features

- Provides trusted components for rapid construction of new parallel PIC codes (You-PICK).
- Provides optimized common algorithms for different computer architectures (work in progress for GPUs).
- Support multiple physics models, levels of accuracy, optimizations.
- Supports both MPI and threaded programming models.
- Hides parallel processing by reusing communication patterns: Physicists only need to know the data layout.
- Components used in wide variety of applications: Plasma Accelerators (QuickPIC), Magnetic Fusion, Space Physics, Cosmology, Quantum Plasmas, Ion Propulsion (DRACO).



QuickPIC: Plasma Accelerators

(C. K. Huang, *et al.*)



DRACO: Ion Propulsion

(J. Wang, et al)

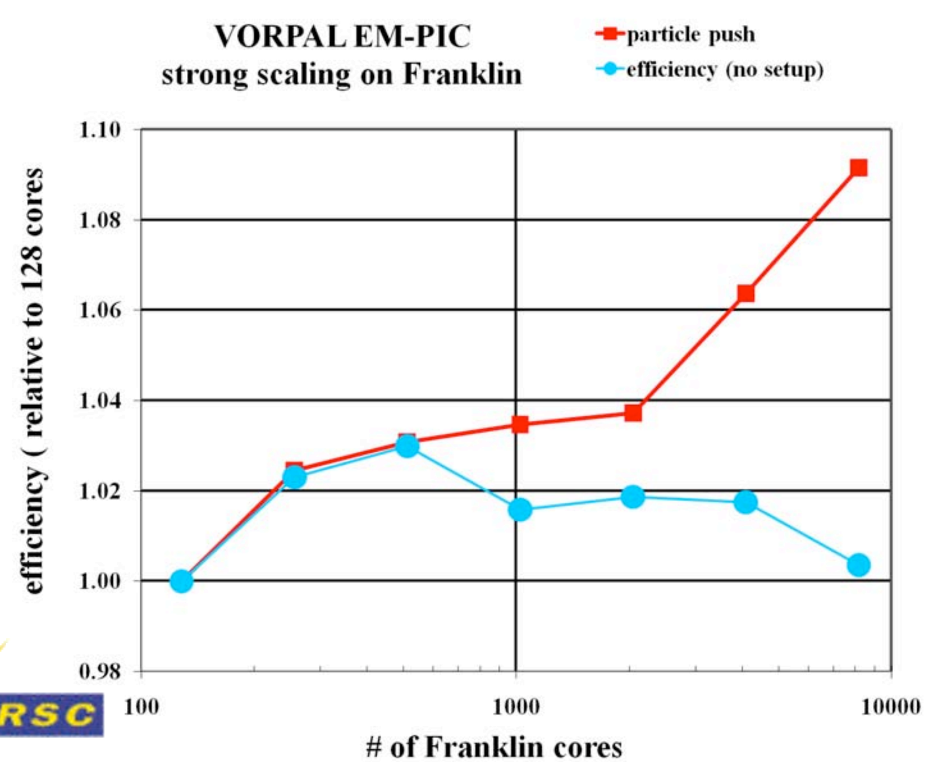
Viktor Decyk,
decyk@physics.ucla.edu
http://exodus.physics.ucla.edu/
http://cfp.ist.utl.pt/golp/epp

*V. K. Decyk, Comp. Phys. Comm. 17, 95 (2007).

Computational VORPAL framework is an important resource for DOE/SC science

VORPAL

- VORPAL is making effective use of supercomputer Franklin
  - 32 active users, spanning HEP, NP, ASCR, FES offices of DOE/SC
  - #6 in hours awarded; #5 in concurrency   (est. by NERSC staff)
  - large production run uses >11k Franklin cores (C. Geddes)
    - Avg. cores per VORPAL run is increasing: from 1,176 (2008) to 2,164 (2009)
  - TB-scale datasets visualized with VisIT, collab. with VACET team



NERSC

Leadership class facility “Franklin” is # 7 on the *Top 500* list;   http://www.top500.org/lists/2008/11

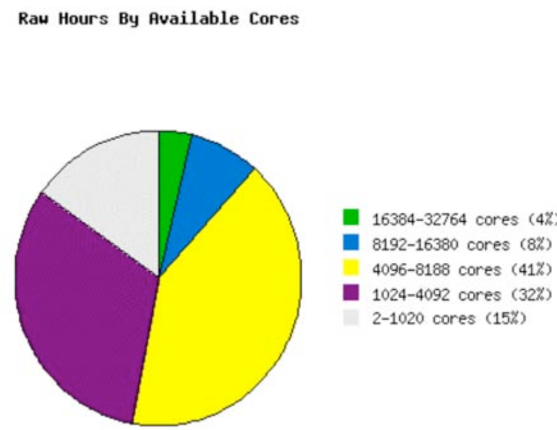
Cray XT4 “Franklin”, from http://www.nersc.gov

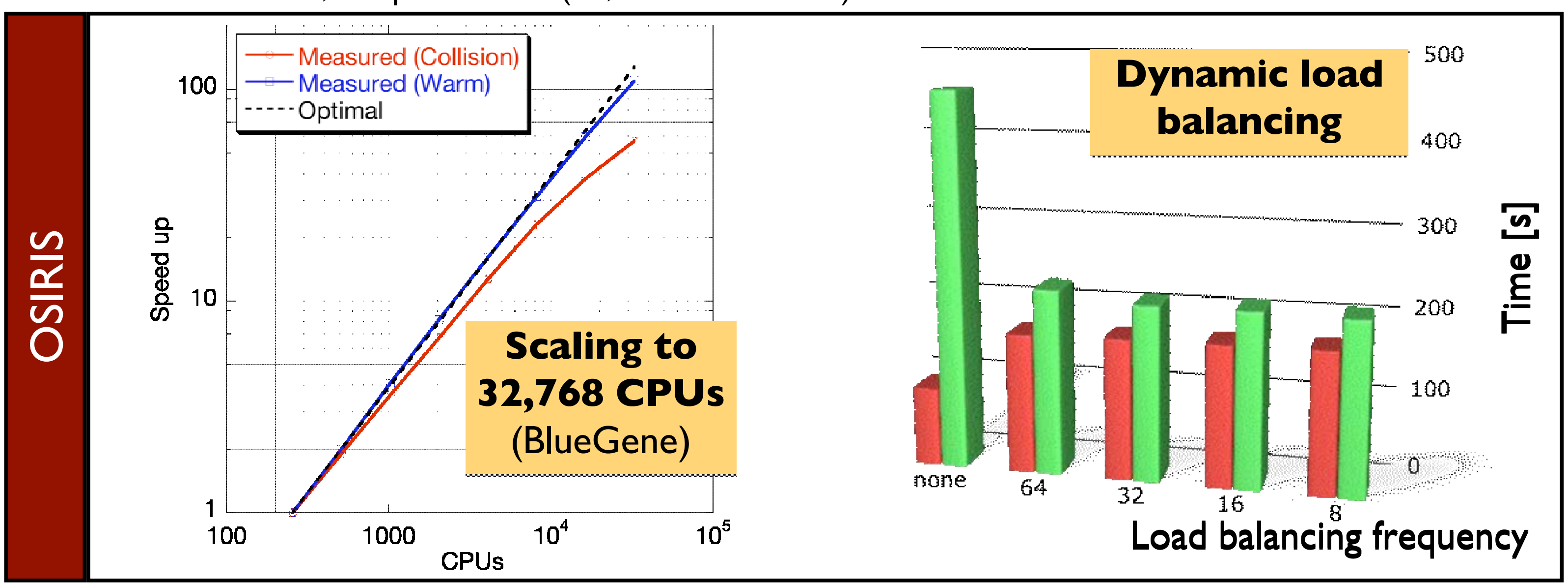
SciDAC codes are extensively used: OSIRIS and QuickPIC

UCLA

Codes make efficient use of computing resources (still working on slide)

- Have been part of 4 INCITE Awards in past two years: OSIRIS is ranked 10th and QuickPIC 15th in usage at NERSC
- OSIRIS is part of 7 projects and QuickPIC is part of 4 projects.
- Optimized single processor performance
  - OSIRIS and UPIC (3D EM):
    - 250 - 300ns/particle/step for push/deposit kernel with linear weighting on AMD processors.
    - Quadratic weighting is only 2 times slower
  - New kernels for UPIC and OSIRIS for the multi-core/GPU/Cell are being developed
- Efficient message passing algorithms leading to strong scaling to domains ~10x10x10 cells
- Dynamic load balancing for particles + fields
- Scales to over 30,000 processors (10,000+ at NERSC)



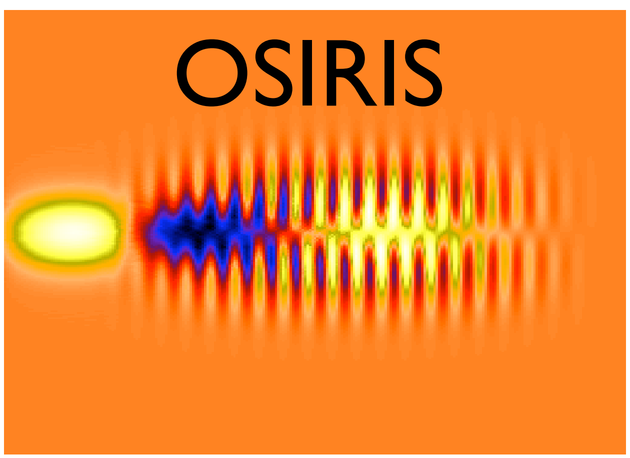


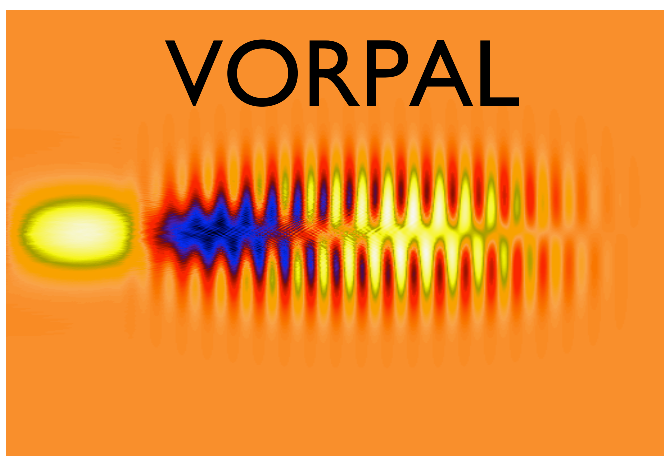
Code and Model Verification

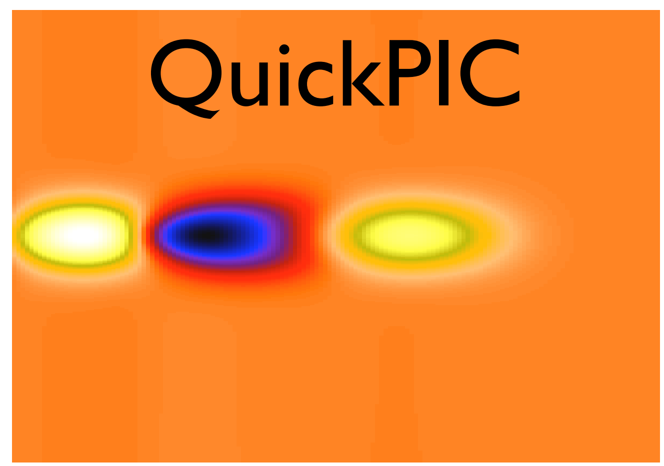
- Code verification is important especially for regime with no analytical results.
- Understanding of different numerical models, algorithms and implementations through benchmarking.

Benchmark Parameters

- λ_0=0.8 μm, I_{peak}~10^{18} W cm^{-2}, τ_{fwhm}=30 fs n_e=1.38×10^{19} cm^{-3}
- 80x80x20 μm^3 box, rectangular mesh of 512x512x512 cells
- 8 particle/cell for full PIC







Recent progress has put plasma-based acceleration at the Forefront of Science

SciDAC codes directly impacted this progress!

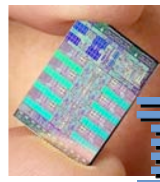
New Advanced Computer Architectures are emerging

SIMD units

Altivec SSE/2/3/4

Cell

PowerXCell



GPUs

CUDA

Multi-core (100 processor) nodes with shared memory is a common feature

But with a variety of implementations
- Cache-based systems (Intel Larrabee)
- Graphical Processors (GPUs)
- Cell Processors (IBM, Sony)

Many of these architectures make use of SIMD processors
- 8-32 processors all executing the same instruction
- IBM Altivec, Cell Processors, Intel SSE, GPUs
- Double precision is often slow, or not supported
- Some systems have cache, some do not

Can one design PIC algorithms which can work on all these architectures?
Important area for future work across project